University of Virginia

Charles L. Brown Department of Electrical and Computer Engineering

Project Proposal: An Innovative High Speed Cache Design

Jacob Breiholz, Leiqing Cai, Ashley Morse, Qing Qin

1. INTRODUCTION

To win the contract for the embedded SRAM application from Portable Instruments Company (PICo), we are now proposing a design of a 64kb high-speed cache.

2. APPROACH

The overarching metric to be optimized is given by:

$$(Active\_Energy\_per\_Access)Delay^2Area(Idle\_Power)$$

Since the metric shows quadratic dependency on both delay and power consumption (though two different kinds of), they are the two majors concerns.

We are planning to take a 5-step approach to realize the goal:

a) Build a toy SRAM structure to get familiar with the basic building blocks of a memory. Assure functional correctness.

b) Look into details of each function block (i.e. decoders, sense-amplifiers, SRAM array), and design them individually to meet our optimization goals.

c) Based on the simple functioning memory, constructs a hierarchical memory.

d) Perform global optimization. For example, insert buffers to reduce delay, and determine buffer sizing using transient simulation results.

e) Add the special feature and make final adjustments.

In particular, step b and d requires building components in different way and using simulation results to make the best possible design decisions. For example, we may test SRAMs with different number of banks (i.e. 1, 2, 4, 8 or 16), and make plots for energy consumption (active/idle) or delay versus bank count, to decide the optimal value for the number of banks.

We have already accomplished step a. Simulation results that verify functionality are available in the Appendix (A2, A3, A4).

3. ARCHITECTURE

3.1 High-Level Architecture

Table 1 – Table 3 specifies the basics parameters, I/O ports assignments, and address partition scheme of the high speed cache we are proposing. A block-level diagram of the memory is available, shown in Figure A1 in the Appendix.

Table 1 Parameters of the High Speed Cache

Parameter	Value
Total Memory Size	64 kbit
Word Size	32 bit
# of Banks	16
# of Columns/Bank	64
# of Rows/Bank	64
# of Words/Row	2

Table 2 Inputs and Outputs

Direction	Pin	Description
Input	CLK	Clock Signal
Input	IN<31:0>	Data Inputs for Write
Input	READ	Read Control Signal
Input	WRITE	Write Control Signal
Input	ADDR<10:0>	Word Address
Output	OUT<31:0>	Data Outputs for Read

Table 3 Partition of the Address Space

Pin	Description
ADDR<10:5>	Row Address
ADDR<4>	Column Address
ADDR<3:0>	Bank Address

3.2 SRAM Bit Cell

In our design, the 6-transistor SRAM bit cell is adopted. Figure 1 shows the schematic with transistor sizes.

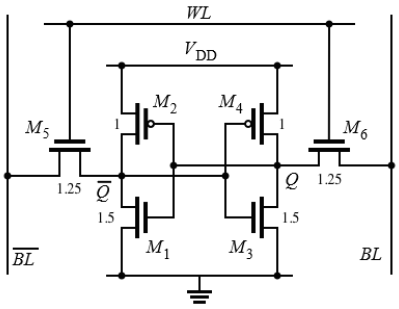

Figure 1 Proposed SRAM Bit Cell with Sizing

To determine the appropriate sizing of the transistors, a few simulations have been conducted. The first experiment aims to gives a rough estimate of the threshold voltage of a characteristic NMOS. To set up, the drain of an NMOS is connected to VDD (1.1V), and the source and bulk are connected to ground. A DC speed from 0 to VDD is performed on the gate terminal. Figure 2 plots the drain current versus the gate voltage. The graph shows the transistor turning-off when the gate voltage goes below roughly 250 mV.

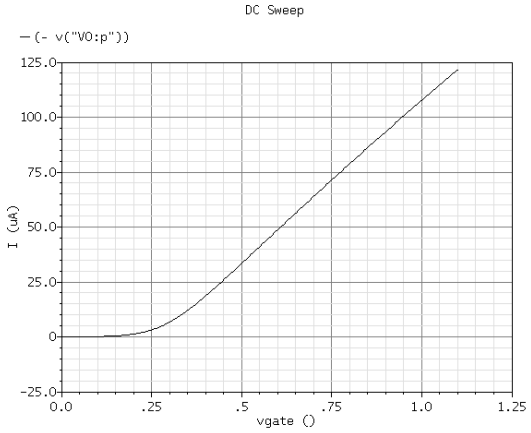

Figure 2 Drain Current vs. Gate Voltage for NMOS

The second simulation aims to figure out the appropriate Cell Ratio (CR) and Pull-up Ratio (PR) for the SRAM bit cell under 45nm technology. Shown in Figure 3, the dotted curve plot the voltage at node Q versus the PR, or W_4/W_6, during a write cycle, in which Q was a 1 and the bit lines try to write a 0. The black curve plots the $V(Q)$ versus the CR, or W_3/W_6, in which Q was a 0 and the bit lines are pre-charged to 1. In the write case, it is desired that $V(Q)$ are pulled lower than V_{TN}, and in the read case, $V(Q)$ should not be pulled higher than V_{TN}. Therefore, we are looking for one point on each curve that sits below the threshold line. We picked 225 mV as the upper bound, shown as the dash horizontal line. We thus picked $W_4/W_6 = 0.8$, $W_3/W_6 = 1.2$, or $W_4 : W_6 : W_3 = 1 : 1.25 : 1.5$

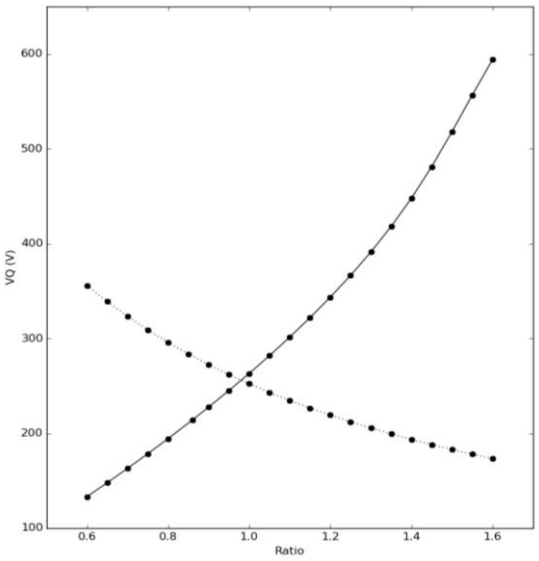

Figure 3 $V(Q)$ vs. CR and PR

To verify that the cell holds data appropriately, a plot of the Static Noise Margin versus p-to-n ratio is necessary, shown in Figure 4. The SNM for the transistor sizing we picked (p-to-n ratio = W_4/W_3 = 0.667) is only 2% less than the maximum value, which is achieved at approximately $W_p/W_n = 1.4$. The result approves our sizing.

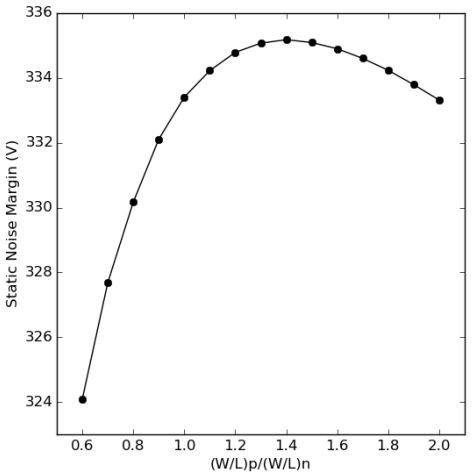

Figure 4 Static Noise Margin for Different *p-to-n* Ratio

3.3 Sense Amplifier

There are two main types of sense amplifiers: voltage mode, and current mode. Voltage mode sense amplifiers work by detecting the voltage differential between the bit lines; current mode sense amplifiers work by detecting the current flowing through the bit lines. We decided to implement a voltage mode sense amplifier for this project, because voltage mode sense amplifiers tend to be faster than current mode sense amplifiers (Mohammad), and our metric for this project emphasizes delay.

There are two main types of voltage mode sense amplifiers: static, and dynamic. Static sense amplifiers read the bit line differential once, and then latch that to the output; dynamic sense amplifiers constantly adjust the output based on the differential between the bit lines. We decided to implement a static design for this project, because dynamic designs require a constant current flow to function, and thus consume a significant amount of power throughout the entire read cycle (Brooks).

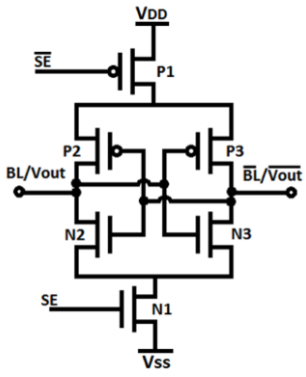

Figure 5 Basic Latch-Based Sense Amplifier

These design decisions led us to the basic latch based sense amplifier, shown in Figure 5. This sense amplifier has two cross-coupled inverters, which utilize positive feedback to latch a value. A drawback of this design is that the bit lines are connected directly to the cross coupled inverters. This means that the sense amplifier has to discharge one of the bit lines before it can latch the output. A common way around this problem is to place pass transistors between the bit lines and sense amplifier, as shown in Figure 6. This decouples the bit lines from the sense amplifier after it is enabled, so it does not have to completely discharge one of the bit lines. This greatly reduces the delay of the sense amplifier.

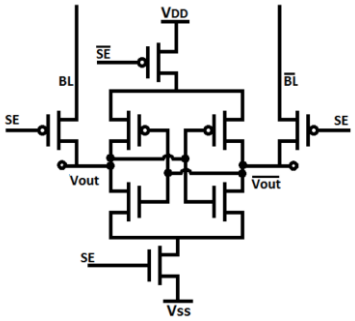

Figure 6 Proposed Sense Amplifier Design

3.4 Decoder

Currently, our design uses a 6to64 row decoder, but if we use banking to reduce delay and size of each block, we could reduce our row decoder to smaller local and global decoders. For now, we will focus only on the 6to64 row and block decoder architecture.

Table 4 Comparing 3 Different Pre-decoder Architectures

	Delay $D_0 \rightarrow Q_{63}$	**Energy**	**# of Transistors**
2to4 Static NOR	102.1 ps	0.715 pJ	572
2to4 Dynamic NOR	61.9 ps	25.1 pJ	560
3to8 Dynamic NOR	42.8 ps	38.4 pJ	460

For the row decoder, we tested several different architectures for delay (from input 0 to output 63), energy, and number of transistors used (Table 4). These follow a 2-stage hierarchy: predecoder stage, then AND the outputs. A predecoder reduces the number of transistors and halves the inputs to the ANDs (Rabaey). Our decoder for Design Review 1 has 3 2to4 static NOR predecoders. Dynamic logic proved to be the better alternative (Rabaey). A decoder with 3 2to4 dynamic NOR predecoders and then with 2 3to8 dynamic NOR predecoders was simulated. Skewing DNOR predecoder also reduces delay (Carr). Figure 7 shows the results, and where 1.4 would be the optimal skewing.

In addition, we will use logical effort, add buffers to reduce parasitic cap., and follow a heuristic (Amrutur) to decrease delay. We will also explore dynamic NAND pre-decoders, because although they are significantly slower than DNOR, they consume much less power (Rabaey). Another option is to look into DRCMOS (Dynamic Range CMOS) to select pre-charging gates and reduce power (Carr). The output of the row decoder will be ANDed with the output of the block decoder to select only one block at a time, further reducing energy (Costanzo). Figure 8 is

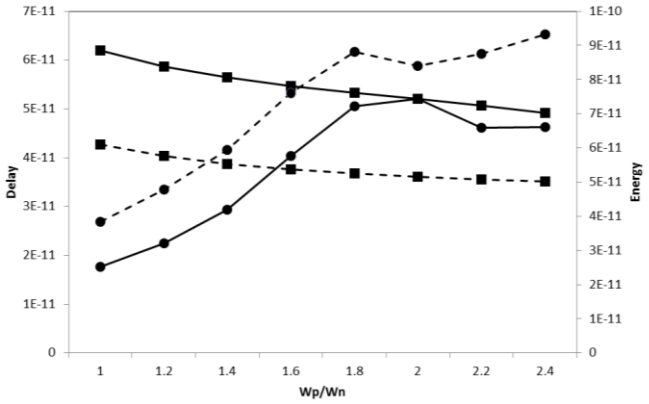

Figure 7 data points of 2to4 DNOR (Delay: solid square, Energy: solid circle) and 3to8 DNOR (Delay: dashed square, Energy: dashed circle)

one possible working design. Simulating idle and active power and reducing it with the above steps and perhaps power gates are the next steps. We expect the outcome to be a decoder that is somewhat fast and reduced in area, and greatly reduced in power.

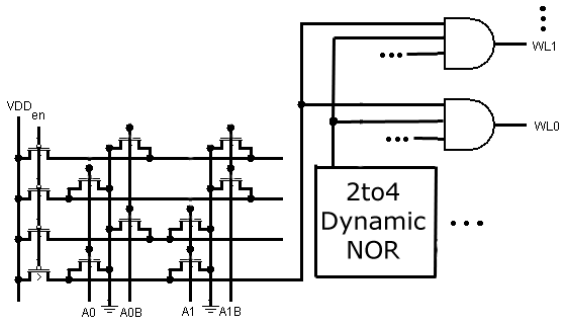

Figure 8 One Possible Row Decoder Schematic

The block decoder will be 3to16, and is planned to be built as a tree-based decoder to reduce transistor count (Rabaey). Although this isn't the fastest implementation, it shouldn't be part of the critical path.

Delay is our most sensitive metric, but we have integrated the decoder into the system such that as long as its delay is less than half a clock period, the delay for accessing the outputs is only 1 TX gate (Figure 9). This drives all outputs to 0 at the beginning.

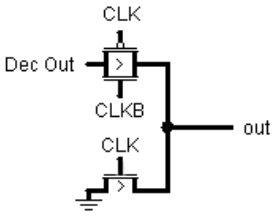

Figure 9 Transmission Schematic after Row Decoder

4. SPECIAL FUNCTIONS

Because delay and power have quadratic effects on the performance metric, we decide to implement the special functions that will reduce the memory access time and both active and idle power. To optimize the delay, we will divide the whole memory into smaller blocks. Shorter word lines and bit lines of each block render smaller capacitance to charge or discharge during read and write operations, and thus result in a shorter delay. The number of blocks we propose is 16 and we will test on different number of blocks to confirm that the option optimizes the delay. To save energy, we will implement lower voltage operations and leakage reduction mode to minimize the active and idle energy.

5. TIMELINE

Week	Milestone
Oct 17 - Oct 23	Block Integration and Optimization
Oct 24 - Oct 30	Modeling and Architecture Optimization
Oct 31 - Nov 6	Entire SRAM Integration
Nov 7 - Nov 13	Low-Voltage Operation
Nov 14 - Nov 20	Leakage Reduction Mode
Nov 21 - Nov 27	Layout and Schematics
Nov 28 - Dec 2	Final Report and Presentations

6. TASK BREAKDOWN

Name	Project Task
Jacob Breiholz	Sense Amplifier Design
	Layout and Schematics
Leiqing Cai	Block Design and Optimization
	Clock and Timing
Ashley Morse	Decoder Design
	Process Corner Simulation
Qing Qin	Architecture
	Overall Simulation

7. REFERENCES

Amrutur, B. S. (1999). *Design and analysis of fast low power SRAMs* (Doctoral dissertation, Stanford University).

Brooks, S., & Cicchetti, A. (2014). *Design of a Low Power Latch Based SRAM Sense Amplifier*. Worcester Polytechnic Institute, Massachusetts.

Carr, D., Park, J., & Reyno, D. (2010). *A High Speed 64kb SRAM Cache in 45nm Technology*. Retrieved from http://venividiwiki.ee.virginia.edu/mediawiki/index.php/Class ECE4332Fall10ProjectTeamXOR

Costanzo, R., Recachinas, M., & Soto, H. (2009). *High Speed64-kb Cache for Mobile Node*. Retrieved from http://venividiwiki.ee.virginia.edu/mediawiki/images/b/b9/EC E4332_CostanzoRecachinasSoto_FinalReport.pdf

Mohammad, B., Dadabhoy, P., Lin, K., & Bassett, P. (2012). Comparative study of current mode and voltage mode sense amplifier used for 28nm SRAM. *Microelectronics (ICM), 2012 24th International Conference on*, pp. 1-6. doi:10.1109/ICM.2012.6471396

Rabaey, J. M., Chandrakasan, A., & Nikolić, B. (2003). *Digital integrated circuits: A Design Perspective* (2nd ed.). Upper Saddle River, N.J.: Pearson Education.